

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
ЗАДАНИЕ.....	3
КОНФИГУРАЦИЯ СИСТЕМ.....	4
ВЫПОЛНЕНИЕ ЗАДАНИЯ.....	6
ВЫВОД.....	10

ЗАДАНИЕ

Написать роль для запуска сервера `nginx`, написать `playbook` для применения роли, провести тестовый запуск `playbook`'а, в случае успешного прохождения теста, применить `playbook` к серверам.

Необходимо добавить переменную, содержащую ФИО, номер группы и номер варианта. Данная переменная должна выводиться в шаблонный файл `nginx`.

Установка пакета выполняется при помощи модуля `APT`, используемого для установки `nginx` в базовой роли.

Добавьте в `playbook task` по установке пакета согласно варианту:

2. `imagemagick`

КОНФИГУРАЦИЯ СИСТЕМ

Создадим две виртуальные машины Deb1 и Deb2, в качестве образа ОС будем использовать Debian, данным машинам присвоим ip адреса 192.168.0.5 и 192.168.0.6 соответственно. Также создадим третью виртуальную машину через которую будет осуществляться работа с Deb1 и Deb2.

На основной машине создадим inventory-файл.

```
root@vbox:~/ansible# cat hosts
[webservers]
server-1 ansible_host=192.168.0.5
server-2 ansible_host=192.168.0.6
[webservers:vars]
ansible_python_interpreter=/usr/bin/python3
```

Рисунок 1 – Содержимое inventory-файла

Проверим подключается ли основная машина к управляемым.

```
root@vbox:~/ansible# ansible -i ./hosts -m ping all
server-1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
server-2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Рисунок 2 – Проверка подключения к подчиненным машинам

Напишем playbook и установим web-сервер Nginx на подчинённые машинки.

```

root@vbox:~/ansible# cat playbook_test
---
- name: Install Nginx to Webservers
  hosts: webservers
  become: yes
  tasks:
    - name: Install Nginx
      apt:
        name: nginx
        state: present
        update_cache: yes
        when: ansible_os_family == "Debian"
        notify: Nginx Systemd
  handlers:
    - name: Nginx Systemd
      systemd:
        name: nginx
        enable: yes
        state: started

```

Рисунок 3 – Текст playbook'a

Выполним этот playbook.

```

root@vbox:~/ansible# ansible-playbook -i hosts playbook_test

PLAY [Install Nginx to Webservers] *****

TASK [Gathering Facts] *****
ok: [server-2]
ok: [server-1]

TASK [Install Nginx] *****
ok: [server-2]
ok: [server-1]

PLAY RECAP *****
server-1      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0     rescued=0    ignored=0
server-2      : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0     rescued=0    ignored=0

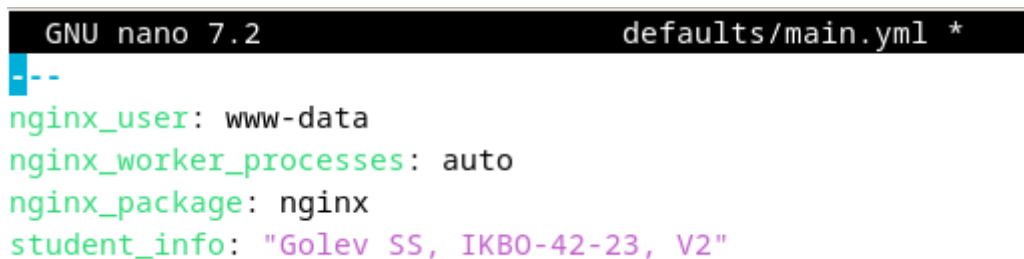
```

Рисунок 4 – Результат выполнения

ВЫПОЛНЕНИЕ ЗАДАНИЯ

Для выполнения персонального задания изменим данные файлы для роли nginx.

В директории defaults изменим файл main.yaml.



```
GNU nano 7.2 defaults/main.yaml *
---
nginx_user: www-data
nginx_worker_processes: auto
nginx_package: nginx
student_info: "Golev SS, IKBO-42-23, V2"
```

Рисунок 5 – Содержание файла main в директории defaults

В директории handlers изменим файл main.yaml.



```
GNU nano 7.2 handlers/main.yaml *
---
- name: restart nginx
  service:
    name: nginx
    state: restarted
```

Рисунок 6 – Содержание файла main в директории handlers

В директории tasks изменим файл main.yaml.

```

GNU nano 7.2                               nginx/tasks/main.yml *
- name: Install Nginx
  apt:
    name: "{{ item }}"
    state: present
    update_cache: yes
  loop:
    - "{{ nginx_package }}"
    - imagemagick

- name: Create config dir
  file:
    path: /etc/nginx/conf.d
    state: directory
    mode: '0755'

- name: Deploy index page
  template:
    src: index.html.j2
    dest: /var/www/html/index.html

- name: Running
  service:
    name: nginx
    state: started
    enabled: yes

```

Рисунок 7 – Содержание файла main в директории tasks

В директории templates создадим html файл и заполним его.

```

GNU nano 7.2                               templates/index.html.j2 *
<!DOCTYPE html>
<html>
<head>
    <title>Test Page</title>
</head>
<body>
    <p>{{ student_info }}</p>
</body>
</html>

```

Рисунок 5 – Содержание файла index.html.j2 в директории templates

Создадим playbook файл.

```

GNU nano 7.2                               nginx_playbook *
---
- name: Nginx
  hosts: all
  become: yes
  roles:
    - nginx

```

Рисунок 8 – Содержание файла main в директории defaults

Запустим сервера на подчинённых машинках и проверим корректность их работы.

```

PLAY [nginx] *****

TASK [Gathering Facts] *****
ok: [server-1]
ok: [server-2]

TASK [nginx : Install Nginx] *****
ok: [server-1] => (item=nginx)
ok: [server-2] => (item=nginx)
ok: [server-1] => (item=imagemagick)
ok: [server-2] => (item=imagemagick)

TASK [nginx : Create config dir] *****
ok: [server-2]
ok: [server-1]

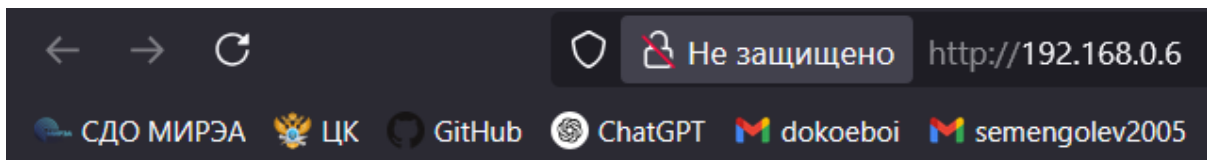
TASK [nginx : Deploy index page] *****
ok: [server-1]
ok: [server-2]

TASK [nginx : Running] *****
ok: [server-2]
ok: [server-1]

PLAY RECAP *****
server-1      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server-2      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

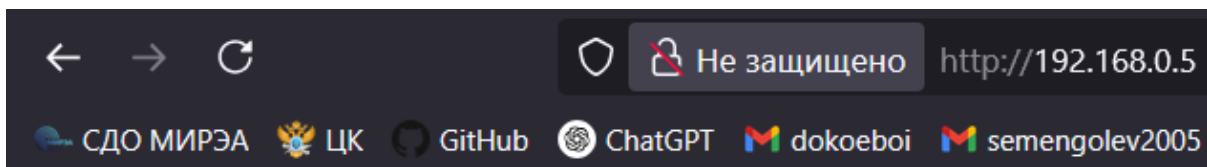
```

Рисунок 9 – Запуск серверов на подчинённых машинках



Golev SS, ИКВО-42-23, V2

Рисунок 10 – Содержимое первого сервера



Golev SS, ИКВО-42-23, V2

Рисунок 10 – Содержимое второго сервера

ВЫВОД

Ansible — мощный инструмент для автоматизации конфигурации и развертывания, использующий простые YAML-файлы (playbook) и модули для управления серверами. В ходе работы были разобраны ключевые аспекты: создание ролей, использование переменных, обработка ошибок. Ansible обеспечивает идемпотентность, что позволяет безопасно повторять выполнение задач. В результате успешно настроен веб-сервер Nginx с выводом пользовательских данных, а также установлены дополнительные пакеты согласно варианту задания.